Car Prices

Charlie's Angels

2024-05-26

```r
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
```

```r
library(readr)
library(readxl)
library(dplyr)
library(ggplot2)
```

```r
carprice <- read_xlsx(path = "CAR PRICE.xlsx", sheet = "Data")
head(carprice)
```

```
## # A tibble: 6 x 26
##   car_ID symboling CarName     fueltype aspiration doornumber carbody drivewheel
##    <dbl>     <dbl> <chr>       <chr>    <chr>      <chr>      <chr>   <chr>
## 1     75         1 buick rega~ gas      std        two        hardtop rwd
## 2     17         0 bmw x5      gas      std        two        sedan   rwd
## 3     74         0 buick cent~ gas      std        four       sedan   rwd
## 4    129         3 porsche bo~ gas      std        two        conver~ rwd
## 5     18         0 bmw x3      gas      std        four       sedan   rwd
## 6     50         0 jaguar xk   gas      std        two        sedan   rwd
## # i 18 more variables: enginelocation <chr>, wheelbase <dbl>, carlength <dbl>,
## #   carwidth <dbl>, carheight <dbl>, curbweight <dbl>, enginetype <chr>,
## #   cylindernumber <chr>, enginesize <dbl>, fuelsystem <chr>, boreratio <dbl>,
## #   stroke <dbl>, compressionratio <dbl>, horsepower <dbl>, peakrpm <dbl>,
## #   citympg <dbl>, highwaympg <dbl>, price <dbl>
```

A. Subset or "split" the carprice into 2 datasets:

- train: contains 150 randomly selected cars from the original dataset
- test: contains the other 55 not selected in the train set. Use 125 as your seed number

```r
set.seed(125)
train_samp <- sample(nrow(carprice), 150)

train <- carprice[train_samp,]
test <- carprice[-train_samp,]

head(train)
```

```
## # A tibble: 6 x 26
```

```
##   car_ID symboling CarName     fueltype aspiration doornumber carbody drivewheel
##    <dbl>     <dbl> <chr>       <chr>    <chr>      <chr>      <chr>   <chr>
## 1      8         1 audi 5000   gas      std        four       wagon   fwd
## 2    152         1 toyota cor~ gas      std        two        hatchb~ fwd
## 3      1         3 alfa-romer~ gas      std        two        conver~ rwd
## 4    203        -1 volvo 244dl gas      std        four       sedan   rwd
## 5     53         1 mazda rx2 ~ gas      std        two        hatchb~ fwd
## 6    163         0 toyota mar~ gas      std        four       sedan   fwd
## # i 18 more variables: enginelocation <chr>, wheelbase <dbl>, carlength <dbl>,
## #   carwidth <dbl>, carheight <dbl>, curbweight <dbl>, enginetype <chr>,
## #   cylindernumber <chr>, enginesize <dbl>, fuelsystem <chr>, boreratio <dbl>,
## #   stroke <dbl>, compressionratio <dbl>, horsepower <dbl>, peakrpm <dbl>,
## #   citympg <dbl>, highwaympg <dbl>, price <dbl>
```

```
head(test)
```

```
## # A tibble: 6 x 26
##   car_ID symboling CarName     fueltype aspiration doornumber carbody drivewheel
##    <dbl>     <dbl> <chr>       <chr>    <chr>      <chr>      <chr>   <chr>
## 1     74         0 buick cent~ gas      std        four       sedan   rwd
## 2     49         0 jaguar xf   gas      std        four       sedan   rwd
## 3    130         1 porsche ca~ gas      std        two        hatchb~ rwd
## 4    205        -1 volvo 264gl gas      turbo      four       sedan   rwd
## 5    106         3 nissan kic~ gas      turbo      two        hatchb~ rwd
## 6    202        -1 volvo 144ea gas      turbo      four       sedan   rwd
## # i 18 more variables: enginelocation <chr>, wheelbase <dbl>, carlength <dbl>,
## #   carwidth <dbl>, carheight <dbl>, curbweight <dbl>, enginetype <chr>,
## #   cylindernumber <chr>, enginesize <dbl>, fuelsystem <chr>, boreratio <dbl>,
## #   stroke <dbl>, compressionratio <dbl>, horsepower <dbl>, peakrpm <dbl>,
## #   citympg <dbl>, highwaympg <dbl>, price <dbl>
```

B. Using the variables you selected in MP1, fit a multiple linear regression model using the train dataset.

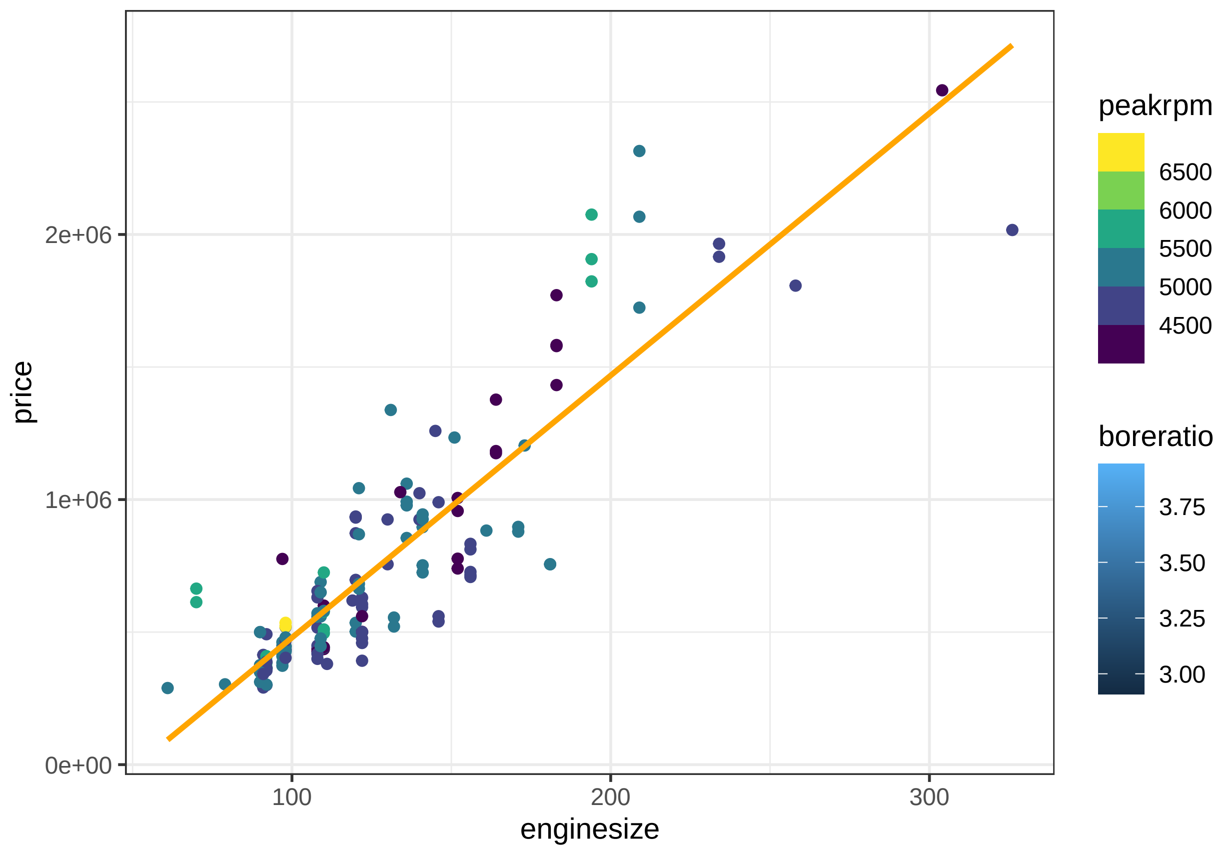
Store the lm class object to an object named model_1. Show results using summary(model_1).

```
model_1 <- lm(price ~ enginesize + peakrpm + boreratio, data = train)
summary(model_1)
```

```
##
## Call:
## lm(formula = price ~ enginesize + peakrpm + boreratio, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -694353  -92401  -27063  100377  708904
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.300e+06  3.495e+05  -3.719 0.000285 ***
## enginesize   9.847e+03  5.312e+02  18.537  < 2e-16 ***
## peakrpm      1.002e+02  3.746e+01   2.674 0.008354 **
```

```
## boreratio    8.479e+04  8.483e+04   1.000 0.319173
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 212100 on 146 degrees of freedom
## Multiple R-squared:  0.7943, Adjusted R-squared:  0.7901
## F-statistic: 187.9 on 3 and 146 DF,  p-value: < 2.2e-16
```

```
ggplot(train, aes(y = price, x = enginesize, color = peakrpm, fill = boreratio)) +
  geom_point() +
  geom_smooth(method = lm, se = F, color = 'orange') +
  scale_color_viridis_b()+
  theme_bw()
```



C. Using model_1, predict the prices in the test dataset.

Store the vector of predicted values in an object named fit_1.

```
fit_1 <- predict(model_1, test)

data_test <- data.frame("Actual price" = test$price, "Predicted price" = fit_1, "Residuals" = test$pric
head(data_test)
```

```
##   Actual.price Predicted.price  Residuals Residuals.Squared
```

```
## 1      2295000       2506069.8 -211069.84       44550477966
## 2      1992000       2024343.5  -32343.53        1046104250
## 3      1760000       1609203.3  150796.71       22739647732
## 4      1268000        950062.4  317937.57      101084298178
## 5      1104000       1294235.6 -190235.63       36189593586
## 6      1067000        940046.1  126953.87       16117284322
```

D. In statistical modelling, the performance is evaluated using some accuracy metrics, such as the Root Mean Square Error (RMSE)

```r
rmse_manual <- sqrt(sum(data_test$Residuals.Squared)/nrow(data_test))
rmse_manual
```

```
## [1] 207745.8
```

```r
rmse_fun <- Metrics::rmse(actual = test$price, predicted = fit_1)
rmse_fun
```

```
## [1] 207745.8
```

Root Mean Square Error = 203229.5